

Predicting quantum evolutions of excitation energy transfer in a light-harvesting complex using multi-optimized recurrent neural networks

Shun-Cai Zhao^{1,2,*} Yi-Meng Huang^{1,2,†} and Zi-Ran Zhao^{1,2}

¹*Department of Physics, Faculty of Science,*

Kunming University of Science and Technology, Kunming, 650500, PR China

²*Center for Quantum Materials and Computational Condensed Matter Physics,*

Faculty of Science, Kunming University of Science and Technology, Kunming, 650500, PR China

(Dated: 17:44, Tuesday 16th January, 2024)

Abstract

Constructing models to discover physics underlying magnanimous data is a traditional strategy in data mining which has been proved to be powerful and successful. In this work, a multi-optimized recurrent neural network (MRNN) is utilized to predict the dynamics of photosynthetic excitation energy transfer (EET) in a light-harvesting complex. The original data set produced by the master equation were trained to forecast the EET evolution. An agreement between our prediction and the theoretical deduction with an accuracy of over 99.26% is found, showing the validity of the proposed MRNN. A time-segment polynomial fitting multiplied by a unit step function results in a striking consistence with analytical formulations for the photosynthetic EET. The work sets up a precedent for accurate EET prediction from large data set by establishing analytical descriptions for physics hidden behind, through minimizing the processing cost during the evolution of weak-coupling EET.

Keywords: Quantum evolution, excitation energy transfer, multi-optimized recurrent neural networks, quantum dynamics

I. INTRODUCTION

Many physical laws were discovered based on data, such as Kepler's three laws via large data of celestial body motion observed by Tycho. Understanding the temporal evolution of photosynthetic excitation energy transfer (EET) in a light-harvesting complex is an important topic of broad interest due to its nearly 100% photosynthetic conversion efficiency, providing an ideal option for mitigating energy crisis[1–3]. Exact numerical simulations of the dynamics of EET in a light-harvesting complex, on the other hand,

requires enormous computational resources[4, 5], which tends to grow exponentially with the number of simulated time steps and system size. Even though many techniques, such as the hierarchy of equations of motion (HEOM) technique[6, 7], multi-configurational time-dependent Hartree (MCTDH)[8], stochastic Liouville-von Neumann equation[9], quasi-adiabatic propagator path-integral (QUAPI)[10], and path-integral Monte Carlo[11] are available, they are inappropriate for examining long-time quantum dynamical evolution.

The current state of a quantum system is mostly determined by and rooted in its early stages of evolution, which enables us to learn long-time

* Corresponding author: zsczhao@126.com

† Co-first author: zscgroup@kmust.edu.cn

evolution of EET in a light-harvesting complex from short-time dynamics without the costly and direct long-time simulations. Once a memory kernel is acquired, the Nakajima-Zwanzig generalized quantum master equation (GQME)[12] provides a broad and formally reliable prescription for achieving this goal[13]. Nonetheless, solving the GQME and directly computing the memory kernel for an arbitrary system is still challenging. The transfer tensor method (TTM) can solve the GQME somehow, but it requires an external numerical methodology to provide a set of dynamical mappings[14–16]. The interplay between machine learning and quantum physics, on the other hand, altered the current situation[17–19] by providing new concepts for modeling the evolution of EET in a light-harvesting complex[20], i.e. intuitively and directly learning from data set without explicit theoretical construction. Artificial neural networks (ANNs)[21] have shown that complex functional dependencies in time series can be learned directly from data[22, 23]. This evades the great efforts to make theoretical analyses, which sometimes become unjustified (e.g., the weak coupling limit) or difficult to derive if just based on phenomenological observations. ANNs have been demonstrated to be capable of solving the master equations governing the dynamics of long-time dissipative open quantum systems[24].

The recurrent neural network (RNN) in particular has an exceptional capacity to interpret an intricate temporal behavior. It can preserve historical information for future purpose of prediction by building a feedback loop that receives both the current stage’s input and the previous step’s output[25]. However, concern of gradient vanishing or exploding behavior due to multiple iterating operations, on the other hand, limits the use of RNNs in long-time scale applications[26]. To address this flaw, this work employs multi-optimized recurrent neural networks (MRNNS) rather than long short-term memory recurrent

neural networks (LSTM-NN)[27, 28], to model the long-term dependencies of the time-series data set, by storing the key information, and predicting the future data that is not currently available. MRNNS are also used as propagators of the time-dependent master equations to regulate the light-harvesting complex over a range of time scales.

The rest of the paper is organized as follows. The quantum processes and the common master equations for photosystem II reaction center (PSII-RC) are described in Sec.2.A, and a sample RNN architecture with optimized hyperparameters is introduced in Sec.2.B. Results are discussed in Sec.3, where we validate the learning model in the interval $[0, 80]$ fs (Sec.3.A) and predict the EET evolution process in the range $[80, 500]$ fs (Sec.3.B), using the polynomial fitting and the analytical expression (Sec.3.C). Finally, conclusions and outlook of the this work are summarized in Sec.4.

II. THEORETICAL MODEL AND MULTI-OPTIMIZED RECURRENT NEURAL NETWORKS

A. Theoretical model for Photosystem II reaction center (PSII-RC)

A typical photosystem II reaction center (PSII-RC) often seen in purple bacteria and oxygen-evolving organisms (cyanobacteria, algae, and higher plants) comprises six pigment molecules located at the core of the complex with two symmetric branches of protein matrix[29]. Four chlorophylls of them (special pair PD_1 , PD_2 and accessory $ChlD_1$, $ChlD_2$) and two pheophytins ($PheD_1$, $PheD_2$), are parallel distributed in these two branches of protein matrix[30]. The pair of chlorophylls, PD_1 and PD_2 , located at the center of the PSII RC act as the primary electron donors, forming two excited states denoted as

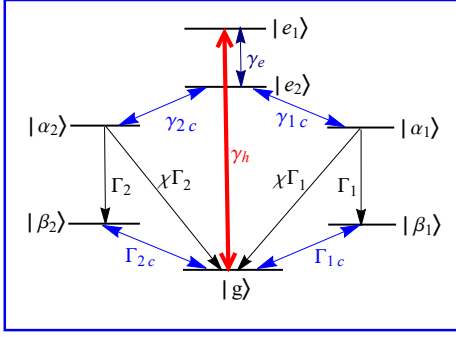


FIG. 1. (Color online) Energy-level framework for the photosynthetic RC with two load-transitions $|\alpha_i\rangle \rightarrow |\beta_i\rangle$, $i=1,2$. The electronic transitions from ground state $|g\rangle$ to two coupled dipoles $|e_1\rangle$ and $|e_2\rangle$ are induced by the high temperature photon bath, while the low temperature phonon bath drives charge transfer from $|e_2\rangle$ to $|\alpha_i\rangle$, $i=1,2$, and $|\beta_i\rangle$, $i=1,2$ to $|g\rangle$ with a termination of electronic circulation.

$|e_{1,2}\rangle$ in Fig.1. Two pheophytin pigments, PheD₁ and PheD₂ couple to the rest of the molecules and act as the electron acceptors $|\alpha_1\rangle$ and $|\alpha_2\rangle$, respectively[31], as shown in Fig.1. This is an energy-level framework abstracted from the PSII-RC[2], where $|\beta_1\rangle(|\beta_2\rangle)$ is a positively charged state after an electron is released and $|g\rangle$ is a ground state.

After absorbing a solar photon, an electron is excited from $|g\rangle$ to $|e_2\rangle$ with a transition rate γ_h , where the excited electron may transit to $|e_1\rangle$ state at a rate γ_e . Then the excited electron is transferred to the acceptors by emitting a phonon via two pathways: $|e_1\rangle(|e_2\rangle) \rightarrow |\alpha_1\rangle$ or $|e_1\rangle(|e_2\rangle) \rightarrow |\alpha_2\rangle$ at emission rates γ_{1c} and γ_{2c} , respectively, with $|\alpha_1\rangle$ and $|\alpha_2\rangle$ being the ion-pair states in these two pathways. Such a process is accompanied by a spatial separation of positive and negative charges induced by the release of the excited

electrons to the plastoquinone molecule, leaving a hole in the dimer. Then the electrons are released from the two acceptors PheD₁ and PheD₂ denoted by $|\alpha_1\rangle \rightarrow |\beta_1\rangle$ (Path 1) and $|\alpha_2\rangle \rightarrow |\beta_2\rangle$ (Path 2) with respective rate $|\Gamma_i\rangle$, $i=1,2$, providing energies for possible work. Finally, the electron returns to the primary electron donor via $|\beta_1\rangle(|\beta_2\rangle) \rightarrow |g\rangle$. As an alternative pathway to the two-step processes $|\alpha_1\rangle(|\alpha_2\rangle) \rightarrow |\beta_1\rangle(|\beta_2\rangle) \rightarrow |g\rangle$, the acceptor-to-donor charge recombination can also be made by directly bringing the system back to the ground state $|g\rangle$ without producing current with rates $\chi\Gamma_{i=1,2}$, where χ is a dimensionless fraction[2] describing the radiative recombination rate of the two pathways.

It is significant to gather a data set, against which the learning model can validate to make prediction. In this work, the data used for training is collected from the initial stage of above PSII-RC, which is generated by the weak contact of the system with the environment.

The unitary evolution of the electron transfer can be described by an electronic Hamiltonian

$$\hat{H}_e = E_g|g\rangle\langle g| + \sum_{i=1}^2 (E_{\alpha_i}|\alpha_i\rangle\langle\alpha_i| + E_{\beta_i}|\beta_i\rangle\langle\beta_i| + E_{e_i}|e_i\rangle\langle e_i|) + (|e_1\rangle\langle e_2| + |e_2\rangle\langle e_1|), \quad (1)$$

via the Lindblad-type master equation

$$\frac{d\hat{\rho}}{dt} = -i[\hat{H}_e, \hat{\rho}] + \mathcal{L}_H\hat{\rho} + \mathcal{L}_{1c}\hat{\rho} + \mathcal{L}_{2c}\hat{\rho} + \mathcal{L}_{3c}\hat{\rho} + \mathcal{L}_\Gamma\hat{\rho} + \mathcal{L}_{\lambda\Gamma}\hat{\rho}, \quad (2)$$

where the strength of the interaction with the environment is comparable with the internal interactions inside the system. The Lindblad-type superoperators in Eq. (2) are listed as

$$\begin{aligned}\mathcal{L}_H\hat{\rho} = & \frac{\gamma_h}{2}[(n_h + 1)(2\hat{\sigma}_{g1}\hat{\rho}\hat{\sigma}_{g1}^\dagger - \hat{\sigma}_{g1}^\dagger\hat{\sigma}_{g1}\hat{\rho} - \hat{\rho}\hat{\sigma}_{g1}^\dagger\hat{\sigma}_{g1}) \\ & + n_h(2\hat{\sigma}_{g1}^\dagger\hat{\rho}\hat{\sigma}_{g1} - \hat{\sigma}_{g1}\hat{\sigma}_{g1}^\dagger\hat{\rho} - \hat{\rho}\hat{\sigma}_{g1}\hat{\sigma}_{g1}^\dagger)],\end{aligned}\quad (3)$$

$$\begin{aligned}\mathcal{L}_{1c}\hat{\rho} = & \sum_{i,j=1}^2 \frac{\gamma_{ijc}}{2}[(n_{1c} + 1)(\hat{\sigma}_{j2}\hat{\rho}\hat{\sigma}_{i2}^\dagger + \hat{\sigma}_{i2}\hat{\rho}\hat{\sigma}_{j2}^\dagger - \hat{\sigma}_{j2}^\dagger\hat{\sigma}_{i2}\hat{\rho} - \hat{\rho}\hat{\sigma}_{i2}^\dagger\hat{\sigma}_{j2}) \\ & + n_{1c}(\hat{\sigma}_{i2}^\dagger\hat{\rho}\hat{\sigma}_{j2} + \hat{\sigma}_{j2}^\dagger\hat{\rho}\hat{\sigma}_{i2} - \hat{\sigma}_{i2}\hat{\sigma}_{j2}^\dagger\hat{\rho} - \hat{\rho}\hat{\sigma}_{j2}\hat{\sigma}_{i2}^\dagger)],\end{aligned}\quad (4)$$

$$\begin{aligned}\mathcal{L}_{2c}\hat{\rho} = & \sum_{i,j=1}^2 \frac{\Gamma_{ijc}}{2}[(n_{2c} + 1)(\hat{\sigma}_{gj}\hat{\rho}\hat{\sigma}_{gi}^\dagger + \hat{\sigma}_{gi}\hat{\rho}\hat{\sigma}_{gj}^\dagger - \hat{\sigma}_{gj}^\dagger\hat{\sigma}_{gi}\hat{\rho} - \hat{\rho}\hat{\sigma}_{gi}^\dagger\hat{\sigma}_{gj}) \\ & + n_{2c}(\hat{\sigma}_{gi}^\dagger\hat{\rho}\hat{\sigma}_{gj} + \hat{\sigma}_{gj}^\dagger\hat{\rho}\hat{\sigma}_{gi} - \hat{\sigma}_{gi}\hat{\sigma}_{gj}^\dagger\hat{\rho} - \hat{\rho}\hat{\sigma}_{gj}\hat{\sigma}_{gi}^\dagger)],\end{aligned}\quad (5)$$

$$\begin{aligned}\mathcal{L}_{3c}\hat{\rho} = & \frac{\gamma_e}{2}[(n_e + 1)(2\hat{\sigma}_{21}\hat{\rho}\hat{\sigma}_{21}^\dagger - \hat{\sigma}_{21}^\dagger\hat{\sigma}_{21}\hat{\rho} - \hat{\rho}\hat{\sigma}_{21}^\dagger\hat{\sigma}_{21}) \\ & + n_e(2\hat{\sigma}_{21}^\dagger\hat{\rho}\hat{\sigma}_{21} - \hat{\sigma}_{21}\hat{\sigma}_{21}^\dagger\hat{\rho} - \hat{\rho}\hat{\sigma}_{21}\hat{\sigma}_{21}^\dagger)],\end{aligned}\quad (6)$$

$$\mathcal{L}_\Gamma\hat{\rho} = \sum_{i=1}^2 \frac{\Gamma_i}{2}(2\hat{\sigma}_{\alpha ii}\hat{\rho}\hat{\sigma}_{\alpha ii} - \hat{\rho}\hat{\sigma}_{\alpha ii} - \hat{\sigma}_{\alpha ii}\hat{\rho}), \quad (7)$$

$$\mathcal{L}_{\chi\Gamma}\hat{\rho} = \sum_{i=1}^2 \frac{\chi\Gamma_i}{2}(2\hat{\sigma}_{bi}\hat{\rho}\hat{\sigma}_{bi}^\dagger - \hat{\rho}\hat{\sigma}_{\alpha ii} - \hat{\sigma}_{\alpha ii}\hat{\rho}). \quad (8)$$

Equation (3) describes the effect of the high temperature reservoirs with n_h denoting its average photon numbers. The low temperature reservoir has the average phonon numbers $n_{1c} = [\exp(\frac{(E_{e2}-E_{\alpha i})}{k_B T_a}) - 1]^{-1}$ in Eq. (4), where $\gamma_{ii}=\gamma_{ic}(\gamma_{jj}=\gamma_{jc})$ are the spontaneous decay rates from level $|e_2\rangle$ to level $|\alpha_i\rangle$ ($i = 1, 2$) respectively. The cross-coupling γ_{ijc} with $\gamma_{ijc}=\gamma_{jic}$ describes the effect of Fano interference. Similarly, another low temperature reservoir is described by Eq. (5) with $n_{2c}=[\exp(\frac{(E_{\beta i}-E_g)}{k_B T_a}) - 1]^{-1}$ being the cold reservoir phonon numbers. Here $\Gamma_{ijc}=\Gamma_{jic}$ is defined as $\Gamma_{ijc} = \eta_1 \sqrt{\Gamma_{ic}\Gamma_{jc}}$ describing the Fano interference induced by the spontaneous decay rates, $\Gamma_{ii}=\Gamma_{ic}(\Gamma_{jj}=\Gamma_{jc})$ ($i, j=1, 2$), from level $|\beta_{i,(i=1,2)}\rangle$ to level $|g\rangle$ with η_1 denoting the quantum interference robustness. In Eq. (6), $n_e=[\exp(\frac{(E_{e1}-E_{e2})}{k_B T_a}) - 1]^{-1}$ is the corresponding thermal occupation numbers of photon at temperature T_a . $\hat{\sigma}_{\alpha ii}=|\alpha_i\rangle\langle\alpha_i|$ ($i=1, 2$) is defined in Eq. (7)

and Eq. (8). Next, 1 million training data sets within 100 fs will be collected from the density matrix element equations (see the Appendix) with parameters[2] listed in Tab. (I).

B. Multi-optimized recurrent neural networks(MRNN)

In this work, the dynamic evolution of populations on each energy level in Fig. 1 was utilized to illustrate EET evolution, and the long-term evolution of EET in the proposed PSII-RC will be predicted by a recurrent neural network (RNN) tuned by four hyperparameters. The simple RNN[21, 32] is a type of neural network that is designed to learn data sequences such as time series, as illustrated in Fig. 2. To further understand how they function, it is reasonable to compare them to more traditional feedforward neural networks. In feedforward neural networks, the input

TABLE I. Model parameters used in the numerical calculations.

| | Values | Units |
|---|-----------------------|-------|
| $E_{e_1} - E_g = E_{e_2} - E_g$ | 0.185 | eV |
| $E_{e_1} - E_{\alpha_1} = E_{e_2} - E_{\alpha_2}$ | 0.2 | eV |
| $E_{\beta_1} - E_g = E_{\beta_2} - E_g$ | 0.2 | eV |
| γ_h | 2.48×10^{-6} | eV |
| γ_e | 0.025 | eV |
| $\gamma_{1c} = \gamma_{2c}$ | 0.012 | eV |
| $\Gamma_1 = \Gamma_2$ | 0.124 | eV |
| $\Gamma_{1c} = \Gamma_{2c}$ | 0.0248 | eV |
| T_a | 0.026 | eV |
| n_h | 6000 | |
| n_e | 0.46 | |
| χ | 0.2 | |
| η_1 | 0.25 | |

data set is propagated step by step via numerous intermediary layers, and training is accomplished by updating the weight matrices as well as the vectors before reaching the final output layer, such that the neural network learns certain desirable input-output correlations. In feedforward neural networks, the input data set is propagated step by step through multiple intermediary levels, and training is performed by updating the weight matrices and vectors before reaching the final output layer, so that the neural network learns some desired input-output relationships behind the data. A feedforward network might theoretically be used to handle temporal data.

However, a feedforward neural network is hardly the best solution since the number of free parameters rapidly grows with the number of time steps. As an alternative option, RNNs handle this issue with a cyclic connection design in which the update rule for the hidden layers at time t is decided not only by the current states S_t , but also by the states at previous times ($t - 1$). $t - 1$, t , $t + 1$ are the time series, x_{t-1} , x_t , x_{t+1} represent

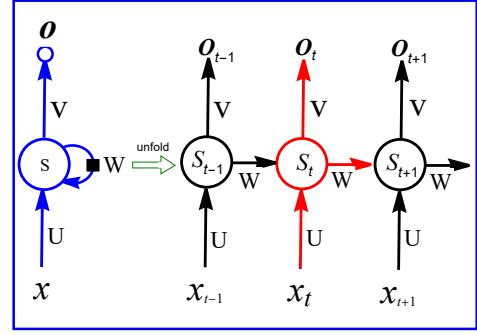


FIG. 2. The architecture of the simple recurrent neural network(RNN) model.

the input sample data, and S_{t-1} , S_t , S_{t+1} represent the memory of the input sample x_i at time t_i , and O_{t-1} , O_t and O_{t+1} , respectively, represent the status information stored at the current time. The RNN update rule is given by

$$S_t = f(Ux_t + WS_{t-1}), \quad (9)$$

$$O_t = g(VS_t). \quad (10)$$

When the data set propagates forward in Fig. 2, the three weight matrices W (input weight), U (input sample weight), and V (output sample weight) do not depend on time t , i.e., weight sharing. The activation functions f and g then take into consideration the free parameters W , U and V , as well as previous memory. The activation function f can be tanh, relu, sigmoid or other functions, while g is usually softmax and other functions. Therefore, the state S_{t-1} of the previous period will participate in the prediction of the state S_t . In other words, the input data and the hidden state of the previous time step will be calculated using the weight matrices so as to generate the hidden state of the current time step[33, 34].

Because of these natural preferences, the Markovian assumption of independent data points at multiple time steps is problematic in the application of RNN. However, if the machine learning model favors noise and minutiae, and ignores the general trends and patterns in training

the data set, it will perform well on trained data but poorly on new data, a phenomenon termed overfitting. In this work, we will use a simple RNN with a multi-optimizer to predict the quantum evolutions of EET in light-harvesting complexes. Additionally, the multi-optimized hyperparameters are addressed further below.

1. Learning rate regulator (LRR)

A learning rate regulator (LRR) will be fitted to the simple RNN, which will determine how much the model parameters are updated in each iteration and how far down the gradient the parameter moves with each update[35, 36]. The exponential attenuation regulator, cosine LRR, pre-heating regulator, and LR attenuation regulator are all commonly used LRRs. A successful LRR should be able to optimize the model while avoiding overfitting and underfitting. If the LRR is set too high or too low, the model may diverge or converge slowly, requiring more training rounds to get the optimal solution.

To simplify the task, we frequently define an LRR at the starting stage of training. During training, the model computes the gradient of the loss function to determine the updating direction of the parameters, whereas the LRR dynamically updates LRR. The performance of the model is evaluated using both the training and verification sets. Because the time series for the quantum evolutions of EET are projected to be long, a faster decay rate LRR, i.e., the exponential decay LRR,

$$LRR = 0.001 \times \exp\left(\frac{-epoch}{10}\right), \quad (11)$$

will be employed to this RNN. Here the *epoch* parameter means the iterations during the training.

2. Early stop function

In addition to the previously stated anti-overfitting strategies, an early stop function[37] is employed to decrease the overfitting. As implied by its name, the early stop technique completes training before the algorithm overfits and obtains the optimal global outcome, resulting in robust generalization performance, as shown by

$$E_{opt}(t) := \min_{t'} \leq E_{va}(t'), \quad (12)$$

$$GL(t) = 100 \times \left[\frac{E_{va}(t)}{E_{opt}(t)} - 1 \right], \quad (13)$$

where $E_{opt}(t)$ is the ideal verification error set as a function of the number of repetitions t , and $GL(t)$ is the generalization loss evaluating the rate at which the generalization error grows in comparison to the previous lowest error. When the generalization error is large, an early stop is preferable since it indicates that the model has been fitted. Such an ending of training is judged by a threshold of $GL(t)$. The early stop technique's halting criterion is classified into three types: the first, second, and third. In this work, the first type of stop rule is employed to determine the loss function and accuracy on the verification set.

3. Regularization

Regularization with the additional penalty terms into the model's loss function is becoming a popular strategy for reducing model overfitting and improving model generalization in machine learning. There are two typical methods of regularization, namely L1 and L2[38, 39], evaluated by two weight parameters w^* and w following and before the update, respectively. They are given

by

$$w_{L1}^* = \underset{w}{\operatorname{argmin}} \{MSE(y, \hat{y}, w) + \frac{\lambda}{2} \sum_{i=1}^n |w_i|\}, \quad (14)$$

$$w_{L2}^* = \underset{w}{\operatorname{argmin}} \{MSE(y, \hat{y}, w) + \frac{\lambda}{2} \sum_{i=1}^n w_i^2\}, \quad (15)$$

where the sum of the absolute values and the

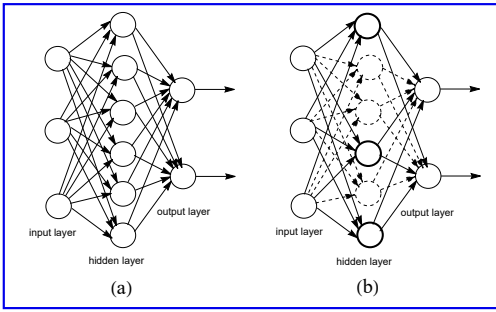


FIG. 3. Recurrent Neural Network without Dropout the structure diagram of standard neural network (a) and its modification with dropout (b).

square of the parameters have been added to the loss function, respectively[39]. The first terms on the right side of the two equations represent the original loss functions, the second terms represent the regularization parameters λ , and the summation is carried out over the numerous model ownership parameters. L1 regularization (14) causes the model to create sparse parameters (some parameters are set to zero), whereas L2 regularization (15) causes the model parameters to be near to but not completely zero. As a result, L1 regularization creates sparse solutions and is more suitable for feature selection, whereas L2 regularization produces a greater number of tiny non-zero weights. In this study, L2 regularization will be used to prevent overfitting in the long-term training data set.

TABLE II. Hyperparameters used in the RNNs training

| | <i>L2</i> | <i>LR</i> | <i>epochs</i> | Dropout |
|---------------------------|-----------|-----------|---------------|---------------------|
| $\rho_{\alpha_1\alpha_1}$ | 0.01 | 0.0001 | 139 | 0.11747474747474748 |
| $\rho_{\alpha_2\alpha_2}$ | 0.01 | 0.0001 | 140 | 0.09779519760654846 |
| $\rho_{e_1e_1}$ | 0.01 | 0.0000001 | 155 | 0.32499999999999996 |
| $\rho_{e_2e_2}$ | 0.01 | 0.0000001 | 125 | 0.32499999999999996 |
| ρ_{bb} | 0.01 | 0.0000001 | 122 | 0.35 |

4. Dropout

In this work, the Dropout regularization technique[40–42] is also applied to the simple RNN. The input training data is carried forward through the neural network, the estimated loss is propagated back, and the parameters are changed using the gradient descent approach, as shown in Fig. 3(a). In Fig. 3(b), the Dropout parameter P is used to deactivate certain neurons with a specific probability during the forward propagation, and the parameters are updated using the gradient descent approach. This method is repeated multiple times so as to properly alleviate overfitting during training in this proposed MRNN.

III. RESULTS AND DISCUSSIONS

A. Training the multi-optimized recurrent neural network (MRNN)

By progressively incorporating a few hyperparameters, such as early stop function, L2 regularization method, LRR, Dropout, and Bayesian optimizer, a multi-optimized RNN model is constructed, which, as a distinguishing feature, utilize the Lindblad-type master equation Eq. (2) to gather data from the proposed PSII-RC, along with certain model parameters listed in Tab. (I). We used Eq. (2) to produce a data set of 1 million

data points in 100 fs, which are divided into training and test sets at 4:1 ratio. The first 800,000 data points serve as the training set, while the remaining 200 000 act as the test set, using the hyperparameters listed in Tab. (II). Fig. 4(a) displays the evolution of the training set fed into this proposed multi-optimized RNN learning model (Original codes in SM1) during the interval $[0, 80]$ fs.

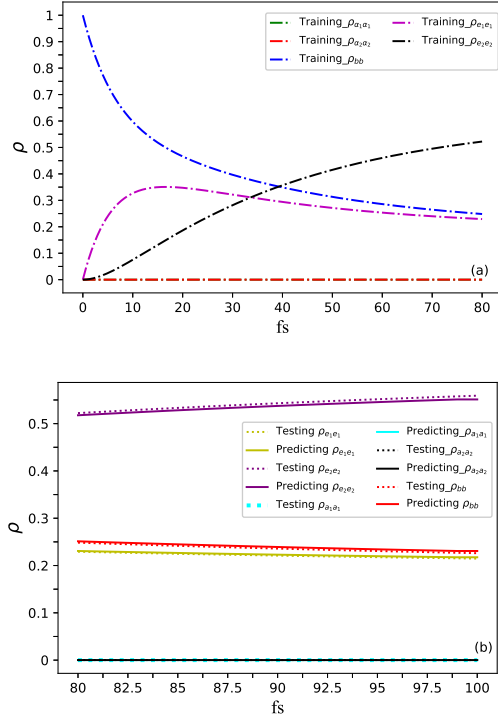


FIG. 4. (Color online) (a) Evolutions of the training population at all excited states within the interval $[0, 80]$ fs. (b) A comparison of learning quality between our proposed learning model and the collected testing set over the range $[80, 100]$ fs. The hyperparameters listed in Tab.(II) and the original codes in SM1, SM2 and SM3 are used for the calculation.

To assess the validity of the proposed learning model, we predict the evolution of EET (solid curves) over a time range from 80 to 100 fs with a comparison to the test set (dotted curves) collected from the aforementioned PSII-RC within the same range, indicating a good agreement between them, as illustrated in Fig. 4(b). This ensures the

high accuracy of this learning model composed of the simple RNN and some optimizers(Original codes in SM1). Supported by these results, the proposed multi-optimized RNN learning model is expected to be able to anticipate the evolution of EET from 80 to 500 fs.

B. EET Predicted by the MRNN

In Fig. 5, the vertical dashed black lines at the 80th fs shows the temporal starting point of our prediction achieved by gradually incorporating hyperparameters into calculation. The cut-off point of 80 fs can be used to evaluate the accuracy of the predictions by checking how precisely the data generated by the prediction coincide with the training data set. After examining Figs. 5(a)-(e), the evolutionary history of each population indicates that our predictions vary against hyperparameters in the following periods. L2 regularization, on the other hand, has little effect on EET prediction, as indicated by the nearly identical curves in Figs. 5(a)-(e). When all the hyperparameters are added into the simple RNN, there is a notable convergence between the training and prediction values at 80 fs, corresponding to the red curves in Figs. 5(a)-(e)(Original codes in SM2).

Consider the prediction of EET on the excited state $|e_1\rangle$ in Fig. 5(a), where the hyperparameters and optimizers are added one by one within the RNN architecture. Three layers with 128, 64, and 32 neurons, respectively, are the essential characteristic neural network architecture. In the absence of optimizers and hyperparameters, the simple RNN timing prediction model fails to yield significant physical results, as shown by the curve “Simple RNN” in the insets of Fig. 5(a), as evidenced by the almost coincided curve with the curve “Simple RNN”. The time forecast provided by the later added early stop function does not satisfy, which can be drawn by the almost

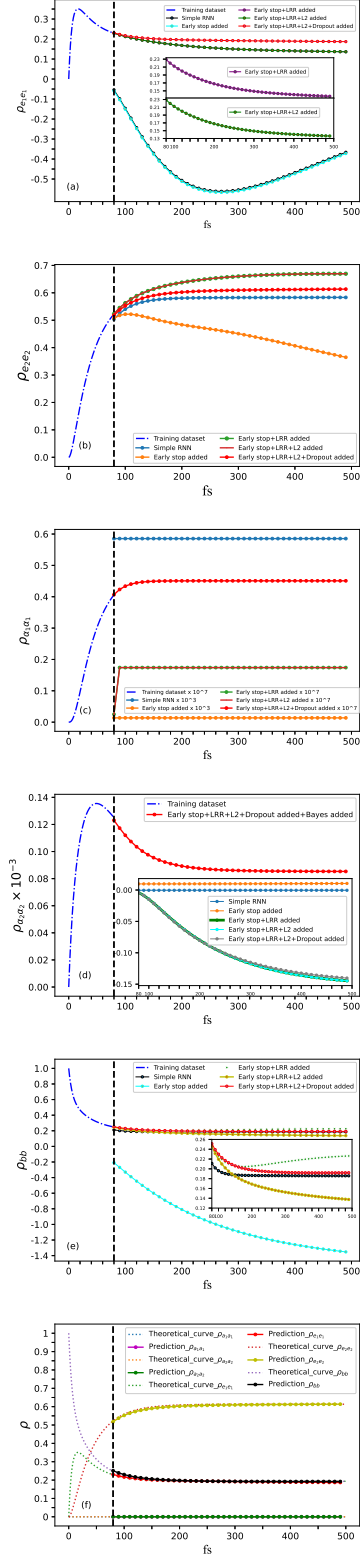


FIG. 5. (Color online) Prediction of the evolutive EET were shown in (a) to (e) via the multi-optimized RNN in the interval [80, 500]fs, and the predictive and theoretical (within [0, 500] fs) EET evolutions were collected in (f) with identical parameters to those in Fig.4.

coinciding curve with the curve “Simple RNN”. Although the apparent physical rule is predicted, their prediction accuracy is practically identical, as shown by the green and purple curves in the inner inset. The evolution curve of $|e_1\rangle$ tends to level off when the Dropout parameter ($P=0.32499999999999996$ is applied for further optimization, as demonstrated by the red curve. The overlapping purple and green curves in the insets of Fig. 5(a) indicates that the development of EET is not sensitive to the L2 regularization on the excited state $|e_1\rangle$.

In Fig. 5(c), the red curve achieves perfect docking with the training data at 80 fs with the Dropout parameter set as $P=0.11747474747474748$ and the initial LR=0.001. As it takes time to manually adjust the hyperparameters of the neural network when optimizing the prediction of the dynamical population on $|\alpha_2\rangle$, the Bayesian optimizer is implemented into the neural network to find the ideal number of layers and neurons. Finally, a neural network composed of 6.23223034545932 layers with 124.19253861421566 neurons per layer was employed, corresponding to the red curve shown in Fig. 5(d). The same neural network architecture parameters are employed to forecast $|b\rangle$, and the inset in Fig. 5(e) clearly shows the roles each hyperparameter plays in predicting the evolution of ρ_{bb} .

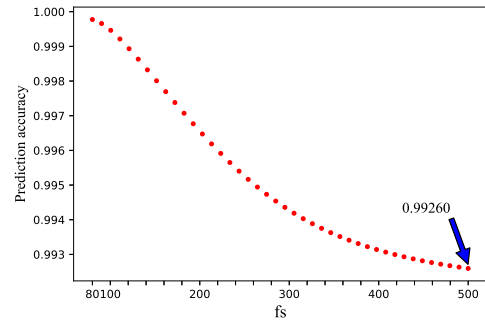


FIG. 6. (Color online) Evolution of prediction accuracy defined by the population sum in the interval [80, 500]fs.

To validate the predictive accuracy of the proposed MRNN in this work, the evolution behaviors of EET are theoretically simulated by using Eq. (2) in the temporal range of $[0, 500]$ fs, as shown by the dotted curves in Fig. 5(f), while the optimal prediction results are re-plotted as curves with points in the time range 80 to 500 fs, as shown by the red curves in Figs. 5(a)-(e) for comparison (Original codes in SM1).

It is found that, as illustrated in Fig. 5(f), the theoretical calculation and the curves predicted by MRNN for photosynthetic EET perfectly coincide, indicating the validity of our proposed MRNN. In addition, the predicted values of each population were added together and divided by one during $[80, 500]$ fs, to evaluate EET predictive precision for the system due to their theoretical sum of one unit. Although the predictive precision decreases against time for this photosynthetic model, it reaches a low of 0.9926 at 500fs, exhibited by an arrow in Fig. 6 (Original codes in SM1). Nevertheless, this accuracy outperforms most RNN learning models, such as those reported by Refs. [43, 44]. At the same time, because our MRNN learning model is born from the simple RNN, long-term prediction inevitably will reveal the inherent flaw of short-term prediction, resulting in a progressive drop in accuracy, as seen by the decreasing feature of Fig. 6.

C. Polynomial fitting and analytical expression for the predictive EET

The purpose of this work is to discover the physical rules underlying photosynthetic EET data. Fitting the predicted findings based on data with a polynomial is an effective mathematical tool to get an analytical understanding of EET. Such a fitting usually relies on the least squares method and lowering the error sum of squares [45, 46]. In Python, the Polynomial Features function is a utility in the scikit-learn library

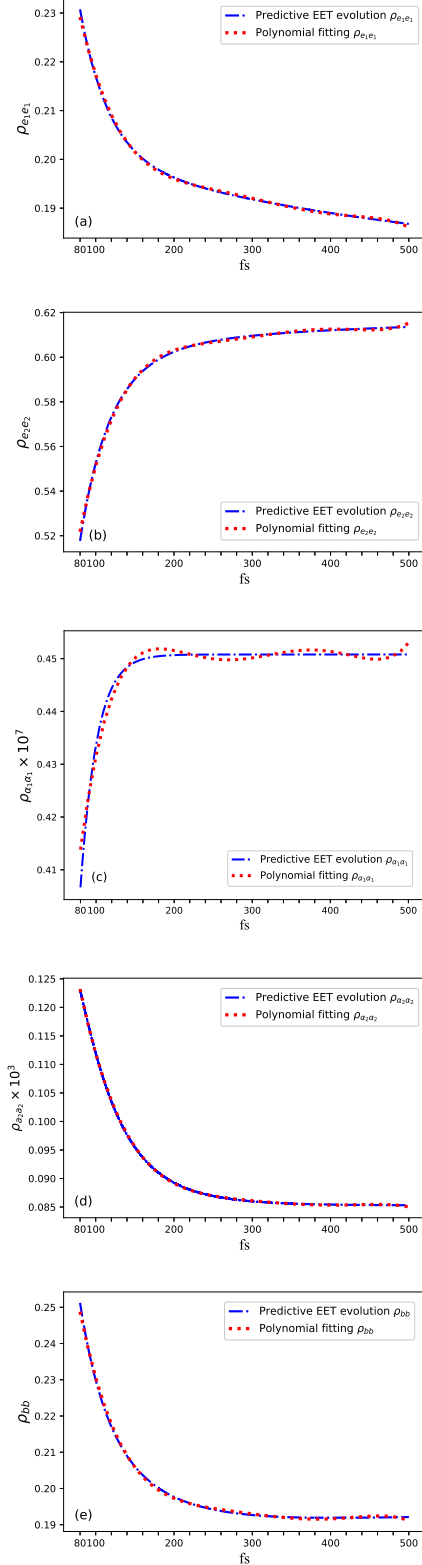


FIG. 7. (Color online) Polynomial fitting (red dotted curves) compared with the predictive evolution (blue dash-dotted curves) of the evolutive EET within the interval of $[80, 500]$ fs.

to generate the polynomial features so as to form a new feature matrix. Finally, the feature matrix is returned for further model training and fitting. When a polynomial of order 2 is provided for a one-dimensional feature A , Polynomial Features generate a new feature matrix containing A to the first and second powers. If the original feature has many dimensions, the resulting feature

matrix will have power combinations for each of them.

Figure 7 displays the differences between the predicted curves and the fitting curves using Polynomial Features in scikit-learn library. The maximum degrees of polynomial fitting for each energy state all are 5 from (a) to (e) in Fig. 7 (Original codes in SM3). And the analytical polynomial fittings to Figs. 7(a)-(e) are also given by

$$\begin{aligned}\rho_{e_1 e_1}(t) = & -0.0022184223655202433t + 1.3936336816327624 \times 10^{-5}t^2 \\ & -4.3658376701560287 \times 10^{-8}t^3 + 6.694506984131638 \times 10^{-11}t^4 \\ & -4.008815848721703 \times 10^{-14}t^5 + 0.33721254828307511, \end{aligned} \quad (16)$$

$$\begin{aligned}\rho_{e_2 e_2}(t) = & 0.0055891350945578t - 3.50638450439065 \times 10^{-5}t^2 \\ & +1.091124267760133 \times 10^{-7}t^3 - 1.6659867620310589 \times 10^{-10}t^4 \\ & +9.953236151938327 \times 10^{-14}t^5 + 0.24964385363485108, \end{aligned} \quad (17)$$

$$\begin{aligned}\rho_{\alpha_1 \alpha_1}(t) = & 4.317228690621229 \times 10^{-10}t - 3.0372753727222955 \times 10^{-12}t^2 \\ & +1.0210132754732106 \times 10^{-14}t^3 - 1.647224363731816 \times 10^{-17}t^4 \\ & +1.0249801765297704 \times 10^{-20}t^5 + 0.00000002169596502, \end{aligned} \quad (18)$$

$$\begin{aligned}\rho_{\alpha_2 \alpha_2}(t) = & -1.8447722039325343 \times 10^{-6}t + 1.0562606143359824 \times 10^{-8}t^2 \\ & -3.040707800147938 \times 10^{-11}t^3 + 4.356557804662399 \times 10^{-14}t^4 \\ & -2.4712186236144212 \times 10^{-17}t^5 + 0.00021705673079966, \end{aligned} \quad (19)$$

$$\begin{aligned}\rho_{bb}(t) = & -0.003319998227995093t + 2.0476938571365984 \times 10^{-5}t^2 \\ & -6.313410066086038 \times 10^{-8}t^3 + 9.606627816067419 \times 10^{-11}t^4 \\ & -5.735689700969715 \times 10^{-14}t^5 + 0.41192131944573884. \end{aligned} \quad (20)$$

The polynomial fitting curves of $\rho_{e_1 e_1}$, $\rho_{e_2 e_2}$, $\rho_{\alpha_2 \alpha_2}$, and ρ_{bb} agree with the predicted curves fairly well, as shown in Fig. 7. In contrast, the fitting result of $\rho_{\alpha_1 \alpha_1}$ is rather poor, as shown in Fig. 7(c). Overfitting is visible in its dynamic population progress. Therefore, an alternative fitting technique should be found for $\rho_{\alpha_1 \alpha_1}$.

Because the characteristic features of an RNN lies in its internal (hidden) loop memory, it is understandable that a dynamic state contains information on all previous input as it evolves

through the data sequence, as shown in Fig. 2. As the polynomial order is increased, the polynomial features over adapt to the past information but perform poorly on fresh data. This enlarges the changing amplitude and change rate that lead to overfitting. An effective method is to derive information based on all previous inputs. For that purpose, a time-segment strategy is implemented, i.e., polynomial fittings in different time ranges

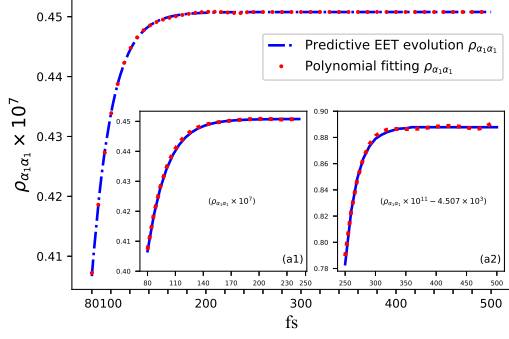


FIG. 8. (Color online) A comparison between the fittings by the time-segment polynomial multiplied by unit step function (red dotted curves) and the predictive evolutions (blue dash-dotted curves) in [80, 500]fs. Time-segment polynomial fittings in comparison with the predictive evolutions shown by the insets(a1) in [80, 250]fs and (a2) in [250, 500]fs.

are carried out and a unit step function

$$f_1(t) = \begin{cases} 1 & t < 250 \\ 0 & t \geq 250 \end{cases}, f_2(t) = \begin{cases} 0 & t < 250 \\ 1 & t \geq 250 \end{cases} \quad (21)$$

is incorporated to the final polynomial expression in each time-segment. The fitting time is divided into two intervals [80, 250] fs and [250, 500] fs, and the polynomial fitting in each interval is multiplied by a unit step function given by Eq. (21). Their final fitting is the sum of the two polynomial fittings, and given by

$$\rho_{\alpha_1\alpha_1}(t_{80-500}) = \rho_{\alpha_1\alpha_1}(t_{80-250}) * f_1(t) + \rho_{\alpha_1\alpha_1}(t_{250-500}) * f_2(t). \quad (22)$$

The curves in Fig. 8 shows the fitting results for the population on the state $|\alpha_1\rangle$ using the aforementioned procedure (Original codes in S-M3). The insets Figs. 8(a1) and (a2), show a comparison between polynomial fitting and predictions in the intervals [80,250]fs and [250,500] fs. The highest degrees of the polynomial fitting is set as 5 in Figs. 8(a1) and (a2). The overlapping between the dotted lines (red) and dash-dotted lines (blue) demonstrates the precision of

this polynomial fitting at various intervals. Furthermore, the analytical functions derived by the time-segment polynomial fitting are, respectively, given by

$$\begin{aligned} \rho_{\alpha_1\alpha_1}(t_{80-250}) &= 1.7388382452293818 * 10^{-9}t \\ &\quad - 1.8538908793974676 * 10^{-11}t^2 \\ &\quad + 9.883817236997048 * 10^{-14}t^3 \\ &\quad - 2.6256355830257586 * 10^{-16}t^4 \\ &\quad + 2.7737661365102853 * 10^{-19}t^5 \quad (23) \\ \rho_{\alpha_1\alpha_1}(t_{250-500}) &= 2.500773638209706 * 10^{-12}t \\ &\quad - 1.2744394675654058 * 10^{-14}t^2 \\ &\quad + 3.227678944102217 * 10^{-17}t^3 \\ &\quad - 4.0616638358969016 * 10^{-20}t^4 \\ &\quad + 2.031479651834943 * 10^{-23}t^5 \quad (24) \end{aligned}$$

The comparison made between the whole period fitting and predictive evolutive curves in [80, 500]fs for the dynamics populations on the states $|\alpha_1\rangle$ demonstrates that the time-segment polynomial fitting can well overcome the overfitting in Fig. 7(c).

Figure (9) depicts the total fitting loss rate compared to the predictive results, a physical quantity assessing the precision of the fitting technique employed in this work. The loss rate curve in Fig. (9) shows an oscillating behavior in the time interval of interest. Although a jumpy loss rate can be seen both at the initial and the final stages of the time interval, it still remains around an order of 10^{-5} , ensuring a high accuracy for this MRNN prediction utilized in this work.

IV. CONCLUSION AND OUTLOOK

In summary, fed by the original PSII-RC data set, a MRNN strategy is proposed to forecast the EET evolution, with an accuracy of over 99.26% within 500fs if compared to the theoretical deduction. The polynomial fitting is also implemented for the EET evolutions so as to get analytical re-

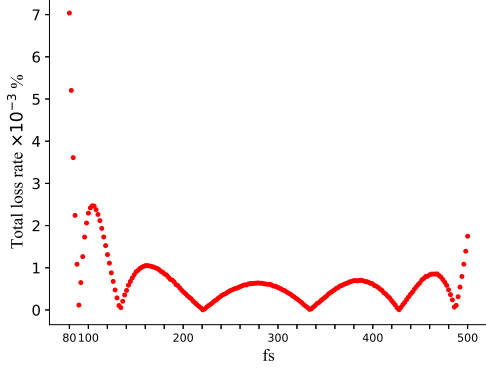


FIG. 9. (Color online) Total loss rate of the polynomial fitting versus the prediction within the range of [80, 500]fs.

sults. The predicted EET evolutions are also subjected to time-segment polynomial fitting multiplied by a unit step function, and the analytical formulations with high precision demonstrate the closeness to physical law. The results reveal that the proposed MRNN is a valid and powerful data mining tool for forecasting the evolution of EET in a light-harvesting complex, and this study establishes a precedent for predicting EET from data using MRNN. A comparison to experimental data in the future is expected to assess

the validity of this learning model.

AUTHOR CONTRIBUTIONS

S. C. Zhao conceived the idea. Y. M. Huang performed the numerical computations and wrote the draft, and S. C. Zhao did the analysis and revised the paper. Z. R. Zhao gave some discussion in the numerical computations.

V. ACKNOWLEDGMENTS

S. C. Zhao is grateful for the significant discussions and language polishing from Prof. Chen-Xu Wu of Xiamen Univ., and for fundings from the National Natural Science Foundation of China (grants 62065009 and 61565008) and Foundation for Personnel training projects of Yunnan Province, China (grant 2016FB009).

VI. APPENDIX

The density matrix dynamic element equations are given by

$$\begin{aligned}
\dot{\rho}_{e_1 e_1} &= -\gamma_e[(n_e + 1)\rho_{e_1 e_1} - n_e \rho_{e_2 e_2}] - \gamma_h[(n_h + 1)\rho_{e_1 e_1} - n_h \rho_{gg}], \\
\dot{\rho}_{e_2 e_2} &= \gamma_e[(n_e + 1)\rho_{e_1 e_1} - n_e \rho_{e_2 e_2}] - \gamma_{1c}[(n_{1c} + 1)\rho_{e_2 e_2} - n_{c1}\rho_{\alpha_1 \alpha_1}] \\
&\quad - \gamma_{2c}[(n_{1c} + 1)\rho_{e_2 e_2} - n_{1c}\rho_{\alpha_2 \alpha_2}] + 2\gamma_{12c}n_{c1}Re[\rho_{\alpha_1 \alpha_2}], \\
\dot{\rho}_{\alpha_1 \alpha_1} &= \gamma_{1c}[(n_{1c} + 1)\rho_{e_2 e_2} - n_{1c}\rho_{\alpha_1 \alpha_1}] - \gamma_{12c}n_{1c}Re[\rho_{\alpha_1 \alpha_2}] - (1 + \lambda)\Gamma_1\rho_{\alpha_1 \alpha_1}, \\
\dot{\rho}_{\alpha_2 \alpha_2} &= \gamma_{1c}[(n_{1c} + 1)\rho_{e_2 e_2} - n_{1c}\rho_{\alpha_1 \alpha_1}] - \gamma_{12c}n_{1c}Re[\rho_{\alpha_1 \alpha_2}] - (1 + \lambda)\Gamma_2\rho_{\alpha_2 \alpha_2}, \\
\dot{\rho}_{\alpha_1 \alpha_2} &= -i\Delta_1\rho_{\alpha_1 \alpha_2} - \frac{1}{2}(\gamma_{1c} + \gamma_{2c})n_{1c}\rho_{\alpha_1 \alpha_2} + \frac{1}{2}\gamma_{12c}[2(n_{1c} + 1)\rho_{e_2 e_2} - n_{1c}\rho_{\alpha_2 \alpha_2} - n_{1c}\rho_{\alpha_1 \alpha_1}] \\
\dot{\rho}_{\beta_1 \beta_1} &= \Gamma_1\rho_{\alpha_1 \alpha_1} - \Gamma_{1c}[(n_{2c} + 1)\rho_{\beta_1 \beta_1} - n_{2c}\rho_{gg}] - \Gamma_{12c}(n_{2c} + 1)Re[\rho_{\beta_1 \beta_2}], \\
\dot{\rho}_{\beta_2 \beta_2} &= \Gamma_2\rho_{\alpha_2 \alpha_2} - \Gamma_{2c}[(n_{2c} + 1)\rho_{\beta_2 \beta_2} - n_{2c}\rho_{gg}] - \Gamma_{12c}(n_{2c} + 1)Re[\rho_{\beta_1 \beta_2}], \\
\dot{\rho}_{\beta_1 \beta_2} &= -i\Delta_2\rho_{\beta_1 \beta_2} - \frac{1}{2}(\Gamma_{1c} + \Gamma_{2c})(n_{2c} + 1)\rho_{\beta_1 \beta_2} - \frac{1}{2}\Gamma_{12c}[(n_{2c} + 1)\rho_{\beta_1 \beta_1} + (n_{2c} + 1)\rho_{\beta_2 \beta_2} - 2n_{2c}\rho_{gg}], \\
\rho_{gg} &= 1 - \rho_{e_1 e_1} - \rho_{e_2 e_2} - \rho_{\alpha_1 \alpha_1} - \rho_{\alpha_2 \alpha_2} - \rho_{\beta_1 \beta_1} - \rho_{\beta_2 \beta_2},
\end{aligned}$$

where $\Delta_1 = E_{\alpha_1} - E_{\alpha_2}$ and $\Delta_2 = E_{\beta_1} - E_{\beta_2}$ are the splitting of the states $|\alpha_1\rangle(|\alpha_2\rangle)$ and $|\beta_1\rangle(|\beta_2\rangle)$.

We utilize the equations to simulate dynamics of

This manuscript has associated data in a data repository. [Authors' comment: All data included in this manuscript are available upon reasonable request by contacting with the corresponding author]. The Supporting Information is available free of charge at: Supplements to MRNN

-
- [1] F. H. Alharbi and S. Kais. Theoretical limits of photovoltaics efficiency and possible improvements by intuitive approaches learned from photosynthesis and quantum coherence. *Renewable & Sustainable Energy Reviews*, 43:1073, 2015.
- [2] L. F. Li and S. C. Zhao. Influence of the coupled-dipoles on photosynthetic performance in a photosynthetic quantum heat engine. *Chin. Phys. B*, 30(4):044215, 2021.
- [3] L. F. Li, S. C. Zhao, and L. X. Xu. Charge-transport enhanced by the quantum entanglement in the photosystem II reaction center. *Europ. Phys. J. Plus*, 136, 10 2021.
- [4] A. Kimura and Y. Fujihashi. Quantitative correction of the rate constant in the improved variational master equation for excitation energy transfer. *J Chem. Phys.*, 141(19), 2014.
- [5] P. F. Huo and T. F. Miller III. Electronic coherence and the kinetics of inter-complex energy transfer in light-harvesting systems. *Phys. Chem. Chem. Phys.*, 17(46):30914, 2015.
- [6] J. Strmpfer and K. Schulten. Open quantum dynamics calculations with the hierarchy equations of motion on parallel computers. *J Chem. Theor. and Comput.*, 8:2808, 2012.
- [7] Y. Tanimura. Numerically exact approach to open quantum dynamics: The hierarchical equations of motion (heom). *J. Chem. Phys.*, 153:020901, 7 2020.
- [8] J. T. Stockburger and H. Grabert. Exact c-number representation of non-markovian quantum dissipation. page 170407.
- [9] H. J. Meyer, U. Manthe, and L. S. Cederbaum. The multi-configurational time-dependent hartree approach. *Chem. Phys. Lett.*, 165:73, 1990.
- [10] N. Makri. Quantum dissipative dynamics: A numerically exact methodology. *J. Phys. Chem. A*, 102:4414, 05 1998.
- [11] D. Kast and J. Ankerhold. Persistence of coherent quantum dynamics at strong dissipation. *Phys. Rev. Lett.*, 110:010402, 2013.
- [12] S. Nakajima. On quantum theory of transport phenomena steady diffusion. *Progress of Theoretical Physics*, (20):948–959, 1958.
- [13] A. Kelly and T. E. Markland. Efficient and accurate surface hopping for long time nonadiabatic quantum dynamics. *J Chem. Phys.*, 139:014104, 2013.
- [14] J. Cerrillo and J. S. Cao. Non-markovian dynamical maps: Numerical processing of open quantum trajectories. *Phys. Rev. Lett.*, 112:110401, 2014.
- [15] A. A. Kananenka, C. Y. Hsieh, J. S. Cao, and E. Geva. Accurate long-time mixed quantum-classical liouville dynamics via the transfer tensor method. *J. Phys. Chem. Lett.*, 7:4809, 2016.
- [16] Y. Q. Chen, K. L. Ma, Y. C. Zheng, J. Allcock, S. Y. Zhang, and C. Y. Hsieh. Non-markovian noise characterization with the transfer tensor method. *Phys. Rev. Appl.*, 13:034045, 2020.

- [17] K. Naicker, I. Sinayskiy, and F. Petruccione. Machine learning for excitation energy transfer dynamics. *Phys. Rev. Research*, 4(3):033175, 2022.
- [18] Y. Han, I. Ali, Z. Wang, J. Cai, and J. Li. Machine learning accelerates quantum mechanics predictions of molecular crystals. *Physics Reports*, 934:1–71, 2021.
- [19] K. N. Lin, J. W. Peng, F. L. Gu, and Z. G. Lan. Automatic evolution of machine-learning-based quantum dynamics with uncertainty analysis. *J Chem. Theor. and Comput.*, 18(10):5837, 2022.
- [20] A. Ullah and P. O. Dral. Predicting the future of excitation energy transfer in light-harvesting complex with artificial intelligence-based quantum dynamics. *Nat. Commun.*, 13(1930), 2022.
- [21] L. Banchi, E. Grant, A. Rocchetto, and S. Severini. Modelling non-markovian quantum processes with recurrent neural networks. *New J. Phys.*, 20(12), 2018.
- [22] S. T. Tsai, E. J. Kuo, and P. Tiwary. Learning molecular dynamics with simple language model built upon long short-term memory neural network. *Nat. Comm.*, 11:5115, 2020.
- [23] L. E. H. Rodriguez and A. A. Kananenka. Convolutional neural networks for long-time dissipative quantum dynamics. *J. Phys. Chem. Lett.*, 12(2476), 2021.
- [24] M. J. Hartmann and G. Carleo. Neural-network approach to dissipative quantum many-body dynamics. *Phys. Rev. Lett.*, 122:250502, 2019.
- [25] D. Lemm, G. F. Von Rudorff, and O. A. Von Lilienfeld. Machine learning based energy-free structure predictions of molecules, transition states, and solids. *Nat. Commun.*, 12:4468, 2021.
- [26] S. Nangia, A. W. Jasper, T. F. Miller III, and D. G. Truhlar. Army ants algorithm for rare event sampling of delocalized nonadiabatic transitions by trajectory surface hopping and the estimation of sampling errors by the bootstrap method (vol 120, 3586, 2004). *J. Chem. Phys.*, 120(8):3586, 2004.
- [27] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9:1735, 1997.
- [28] K. N. Lin, J. W. Peng, F. L. Gu, and Z. G. Lan. Simulation of open quantum dynamics with bootstrap-based long short-term memory recurrent neural network. *J. Phys. Chem. Lett.*, 12(41):10225, 2021.
- [29] V. I. Novoderezhkin, J. P. Dekker, and R. Van Grondelle. Mixing of exciton and charge-transfer states in photosystem ii reaction centers: Modeling of stark spectra with modified redfield theory. *Biophysical Journal*, 93(4):1293, 2007.
- [30] R. V. Elisabet, I. H. M. Stokkum, V. I. Novoderezhkin, J. P. Dekker, and R. V. Grondelle. Two different charge separation pathways in photosystem II. *Biochemistry*, 49(20):4300, 2010.
- [31] V. I. Novoderezhkin, E. Romero, J. P. Dekker, and R. V. Grondelle. Multiple charge-separation pathways in photosystem ii: modeling of transient absorption kinetics. *Chemphyschem A European Journal of Chemical Physics and Physical Chemistry*, 12(3):681, 2015.
- [32] Y. Qu, M. Zhou, E. K., N. F. Yu, and Z. F. Yu. Resonance for analog recurrent neural network. *ACS Photonics*, 9:1647, 04 2022.
- [33] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673, 1997.
- [34] J. Bueno, S. Maktoobi, L. Froehly, I. Fischer, M. Jacquot, L. Larger, and D. Brunner. Reinforcement learning in a large scale photonic recurrent neural network. *Optica*, 5(6):756, 2018.
- [35] S. H. Ning and K. N. Du. Research on intelligent fault diagnosis of rolling bearing based on adaptive resource allocation deep neural network. *IEEE Access*, 10:62920, 2022.

- [36] Y. Zhang and L. L. Shen. Automatic learning rate adaption for memristive deep learning systems. *IEEE Transactions on Neural Networks and Learning Systems*, (2):1–12, 2023.
- [37] Á. L. Muñoz Castañeda, N. DeCastro-García, and D. Escudero García. Rhoaso: An early stop hyper-parameter optimization algorithm. *Mathematics*, 9(18):2334, 2021.
- [38] D. Krueger, T. Maharaj, J. Kramár, M. Pezeshki, N. Ballas, N. R. Ke, A. Goyal, Y. Bengio, A. Courville, and C. Pal. Zoneout: Regularizing rnns by randomly preserving hidden activations. *arXiv preprint arXiv:1606.01305*, 2016.
- [39] S. Merity, B. McCann, and R. Socher. Revisiting activation regularization for language rnns. *arXiv preprint arXiv:1708.01009*, 2017.
- [40] A. Gajbhiye, S. Jaf, N. A. Moubayed, A. S. McGough, and S. Bradley. An exploration of dropout with rnns for natural language inference. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, (Rhodes, Greece, October 4-7, 2018, Proceedings, Part III 27)*, page 157. Springer, 2018.
- [41] A. Sarma, S. Singh, H. Jiang, R. Zhang, M. T. Kandemir, and C. Das. Structured in space, randomized in time: Leveraging dropout in rnns for efficient training. *Advances in Neural Information Processing Systems*, 34:24545, 2021.
- [42] I. Salehin and D. K. Kang. A review on dropout regularization approaches for deep neural networks within the scholarly domain. *Electronics*, 12(14):3106, 2023.
- [43] B. Adaandm and M. Florescu. Modelling non-markovian dynamics in photonic crystals with recurrent neural networks. *Optical Mater. Express*, 11(7).
- [44] Y. Li, Z. Wang, R. Han, S. Shi, J. Li, R. Shang, H. Zheng, G. Zhong, and Y. Gu. Quantum recurrent neural networks for sequential learning. *Neural Networks*, 166:148–161, 2023.
- [45] J. S. Pei, J. P. Wright, and A. W. Smyth. Mapping polynomial fitting into feedforward neural networks for modeling nonlinear dynamic systems and beyond. *Computer Methods in Applied Mechanics and Engineering*, 194(42):4481, 2005.
- [46] B. A. Escobedo-Trujillo, D. Colorado, W. Rivera, and F. A. Alaffita-Hernández. Neural network and polynomial model to improve the coefficient of performance prediction for solar intermittent refrigeration system. *Solar energy*, 129:28, 2016.